PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04N 7/58	A1	(11) International Publication Number: WO 97/15149 (43) International Publication Date: 24 April 1997 (24.04.97)

(21) International Application Number: PCT/IB96/01074

(22) International Filing Date: 11 October 1996 (11.10.96)

(30) Priority Data:
95202811.6 18 October 1995 (18.10.95) EP
(34) Countries for which the regional or international application was filed: AT et al.

(71) Applicant: PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(71) Applicant *(for SE only)*: PHILIPS NORDEN AB [SE/SE]; Kottbygatan 7, Kista, S-164 85 Stockholm (SE).

(72) Inventors: ELBERS, Gerard, J.; Donk 154, NL-4824 CZ Breda (NL). BOUFFIOUX, Alain, R., P.; Froidbermont 8B, B-4877 Olne (BE).

(74) Agent: STRIJLAND, Wilfred; Internationaal Octrooibureau B.V., P.O. Box 220, NL-5600 AE Eindhoven (NL).

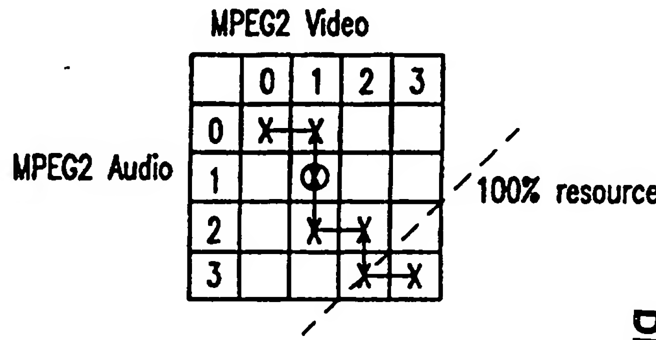
(81) Designated States: CN, JP, KR, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD FOR MAKING A MULTIMEDIA APPLICATION EXECUTABLE ON HARDWARE PLATFORMS WITH VARIOUS DIFFERENT RESOURCE LEVELS, A PHYSICAL RECORD CONTAINING SUCH APPLICATION, AND AN APPARATUS FOR EXECUTING SUCH APPLICATION

(57) Abstract

Multimedia application information is stored for later executability on different platforms, such as Interactive Multimedia Distribution Format (IMDF) complying platforms. The application comprises a plurality of parallel information streams. Execution scalability is enabled by assigning to the streams a single string of quality level items that along the string indicate stepwise increasing quality levels. Each item specifies an associated decoding quality level for a particular stream.



BEST AVAILABLE COPY

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia
AT	Austria
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CS	Czechoslovakia
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgystan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakhstan
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam

Method for making a multimedia application executable on hardware platforms with various different resource levels, a physical record containing such application, and an apparatus for executing such application.

DEFINITIONS

The following definitions are given for better understanding:

Resource	quantitative and/or qualitative properties of the platform such as MIPS of the CPU, memory size in Mbytes, bandwidth of bus in Mbits/sec.
Resource requirement	necessary amount of such resources
Performance	speed of a computer or CPU
Quality level	level of functioning of a decoder, which has been defined independently of the platform

BACKGROUND TO THE INVENTION

The invention relates to a method for storing multimedia application information for execution on different platforms, said information comprising a plurality of information streams for presentation in parallel to a user. Multimedia application programs or applications should be executable on different platforms. Such platform must satisfy minimum requirements in terms of resources, for being able to guarantee that the application in question runs thereon correctly. In various cases the platform has more resources than the minimum. Advantageously, the application should be made scalable, to attain optimum quality level, given the resources of the actually available platform.

SUMMARY TO THE INVENTION

In consequence, amongst other things it is an object of the present invention to provide a method for storing multimedia application information, wherein such scalability is realized in a straightforward manner. Now, according to one of its aspects, the invention is characterized in that for enabling execution scalability of said application, said method collectively assigns to said streams a single string of quality level items that along said string indicate stepwise increasing quality levels, in that each item specifies an associated decoding quality level for a particular one of said streams. Their relative positions in the string give the priorities among the various streams, and also between the various quality

levels for each single stream. Now first, for all streams together, specified minimum platform resources are necessary for the platform to executing the application or not. Moreover, if additional resources are present, assigning thereof to the various streams on the basis of relative priorities is straightforward. This organization allows the platform to easily choose optimum quality level with respect to available resources.

By itself, multimedia executability on different platforms has been considered in EP A 731 971, corresponding US Patent Application 08/533,030 (PHN 15058) to the present assignee, and herein incorporated by reference. In the reference, the multimedia application itself has a multiplatform interpreter, and each actual platform is an instance of an abstract machine. The present invention may be used in such more complex environment, but it is already useful if platform resources differ only in quantitative performance.

In the organization according to the invention, the platform or player knows the resource requirements for each quality level of each decoder. These have been stored in the resource requirement table of the platform (cf. Figure 5). Neither the data streams, nor the application itself need to know these, because generally, at configuring the data streams and writing the application, the target platform is not known yet. The data streams comprise the input data of all associated decoders. In general, no further information is required for realizing the organization of the present invention. The application comprises a model, realized as a real or notional string of items, which describes the incremental model or preference path of the application in view of respective decoder levels.

The resource requirements of a particular platform are fixed for each decoder and each associated quality level when configuring the platform. These are stored in the platform. The producing of the application implies the joining of the data streams in a real-time file, using the so-called discbuilder. Furthermore, an incremental model of the application is built in the application, with one or more calls thereto. Generally, this model is separate from the data streams.

Advantageously, at least one such stream is associated to at least two non-contiguous items of said string. This means that when stream A gets a particular quality level, first the quality of stream B will be improved, before further improvement of stream A can be allowed.

Advantageously, the quality level items are assigned to said streams in a dynamic manner along said application when stored. The writer of an application can thereby decide that the preference amongst the various decoders may vary during the executing of the application. During the execution proper on a particular platform, this may influence the

quality levels actually used, or not.

The invention also relates to a unitary physical record carrier containing such application and to a platform system for playing such multimedia application information, wherein the platform will automatically select appropriate quality levels for the various streams. Further advantageous aspects of the invention are recited in dependent Claims.

BRIEF DESCRIPTION OF THE DRAWING

These and further aspects and advantages of the invention will be described more in detail with reference to preferred embodiments disclosed hereinafter, and in particular with reference to the appended Figures that show:

Figure 1 a flow chart for storing an application according to the invention;

Figure 2 a unitary record carrier so formed;

Figure 3 a platform system for playing the record;

Figure 4 a quality of service management diagram;

Figure 5 a platform resource requirement table;

Figure 6 a preference path diagram.

GENERAL CONSIDERATIONS

In its most complete form, a multimedia and in particular, an IMDF application may contain two types of processing:

- Stream Processing (SP) of audio-visual information streams, such as Video (MPEG), 2D- or 3D-graphics, audio (MPEG, MIDI, PCM, ADPCM), which streams must be processed in a time-conforming manner at the normal rendering speed a human user would expect.
- Non-Stream processing (NSP) that is not directly related to such time-conforming audio-visual streams. For simplicity of wording, the informations relating thereto on the physical record carrier are nevertheless also called streams. This processing type can be subdivided into:
- (a) Real-time constrained NSP, such as simulation calculations, e.g. a flight simulator
- (b) Non-real-time constrained NSP, such as a search in a data base or encyclopedia, for presenting stationary information.

At its low end, each application requires a platform with minimum resources; these may be defined in an applicable standard. On such minimum platform only a limited number of information streams and computations can be processed in parallel. A more powerful platform will in principle allow for a higher quality level compared to the mini-

mum. Examples of such resources are the recurrence frequency of CPU processing cycles, memory size, bus transfer capacity, and I/O-facilities. In certain circumstances the MIPS figure indicating the raw instruction processing speed may need recalculation to MOPS, that is the operation speed as pertaining to the stream in question. Also operational overhead, such as for loading of the whole or part of the application, may need to be taken into account. In case of a server or other multi-process environment, available resources may have to be shared among various applications that run concurrently.

For stream processing (SP), the resource management is done in three steps:
- defining objective quality levels for all feasible decoders;
- selecting an actual quality level for a particular decoder;
- controlling decoder resources at a selected quality level. Herein, decoders in fact are decoder programs that run in parallel on hardware that may be shared or not among various decoders.

In a first approach, non-stream processing (NSP) is considered to be fully determined by the application and non-scalable, as the platform cannot foresee what the NSP will be doing. This means that for NSP resource requirements must be known beforehand in the platform. The amount of allocated resources must be sufficient so that the application will always meet its deadlines if real-time constrained. If not real-time constrained, the allocated resources should be sufficient to avoid any long-term build-up of unfinished business. This generally means that the resources for NSP must be assigned as a whole. Subsequently, any remaining resources can be assigned to SP.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Figure 1 shows a flow chart for storing an application according to the invention. The operations according to this flow chart can be executed directly after the general writing of the application. In block 20 the unprocessed information of the application is received, as pertaining to a particular time interval or scene. Generally, the scene size has been determined by the developer of the application. In so doing, the scene has been usually built up from things like music or movie intervals, stills, graphic elements and others. In the information, the data streams have usually been joined to an interleaved and multiplexed data file. Sufficient processing power is available for executing the application at its highest quality level. During execution of the application, often the processing is linear in time, in that the various streams are presented to a user in mutual synchronism, such as in the way of a movie. However, this behaviour need not be uniform in time, where at one time an image

may freeze, while accompanying music goes on. In an interactive situation, the relative behaviours may be still more non-uniform. In block 22 the first unprocessed item of the application is accessed, in view of therefor determining its string of quality level items.

In block 28, the minimum quality level for executing the (part of the) application is assessed. This decision may be taken by the writer of the application. Next, the scene heading is now provided the minimally required quality levels of all decoders as a first item of the string of quality level items. Inasmuch as this first item may pertain to more than one decoder, it may be a composite item. For being able to attain automatic scalability, subsequent items of the string must now be arranged so that a stepwise increasing quality can be indicated by going along the items of the string. The operation in block 28 can be done either with the isolated scene, or in the context of preceding and/or succeeding scenes. Next, in block 30, exactly one of the streams which allows a plurality of different quality levels is taken at the next higher quality level, and the scene is executed again. This is repeated for each the streams that have incrementable quality levels, which is symbolized by the looping line 31. When the scene has been executed with all possibilities for single-quality increment, in block 32 the best performance thereamongst is chosen. This is generally a subjective action that is done by a designer person. This determines the next item of the string. As long as further quality improvement is possible (34), the system next goes back to block 30. If all streams have been executed at their respective top quality levels, the application scene is loaded (36). The ultimate result of the invention is a unitary carrier that is manufactured in standard manner for media of this kind. In block 38 the processing system is relinquished. For simplicity, only the loading of a single scene has been considered. However, one may decide to execute the flow chart of Figure 1 again for the next scene.

Figure 2 symbolizes a unitary record carrier so formed. The carrier contains eight streams shown as separate columns. For physical storage, generally a serializing operation is effected, such as for Compact-Disc Interactive; this can produce a real-time multiplexed file. Each streams accommodates a block 32 that contains an identifier, length, and various other references. The parallel character of the streams resides in the fact that they can be accessed for replay in an interleaved manner. For brevity, actual organization of physical storage has not been addressed. An exemplary solution has been disclosed in copending EP Patent Application 95202623.5, corresponding United States Patent Application 08/...,... (PHN 15480) to the present assignee and herein incorporated by reference. In such interleaving, the player keeps track of the stream that is actually being accessed. As recited supra, the string of quality level items which indicates the stepwise increasing quality

levels is stored in a scene header indicated by X in the Figure.

Figure 3 shows a platform system for playing the record carrier. The disc-shaped record 40 is accessed by read arrangement 42 under control of positioning signals on line 44. The player proper 60 has user interface 50 for inputting user commands and detector block 46 for detecting the string of quality level items recited supra. This block also knows the required resources for the various quality levels of the various decoders, as pertaining to the platform in question. Therefore it may select the actually attainable level of quality, in that certain requirements can be met, but others would overtax the platform. This will hereinafter be discussed further. Central processing device 48 processes the streams in question in accordance with the above selection. Such processing may further be under control by user commands on interface 50. In principle, the user may overrule the quality levels selected by the platform. For example, a user may prefer audio representation above video, and therefor by preference allocate more resources to audio processing than according to the relative priorities assigned thereto by the designer of the application. Device 48 outputs various user information signals to various user output facilities 52, 54, 56, 58, that may contain audio channels, video channels, and data channels. Various ones of these may be collocated, such as the displaying of video and animation in parallel on the same screen. The processing also causes positioning on line 44 for read arrangement 42.

Hereinafter, resource management will be discussed more in detail. The processing device 48 includes various facilities, such as memory, bus, and I/O features. The information thereof is known to detector block 46, which also knows the amounts of resources necessary for the various quality levels. This allows device 46 to choose an optimum quality level that can be realized with the resources actually present, as pertaining to the actual application. If an appropriate level of performance has been chosen, the necessary resources are allocated, and the application may be executed. If the resources are however insufficient such as defined in terms of bandwidth, delay, error rate, or other, the player will inform the user and the execution of the application will remain blocked. Upon meeting a subsequent scene, the resource allocation may have to recommence. If the resource requirements are dynamically varying along the application, the application may signal minimum overall quality level required, so that the overall executability of the application may be decided by the platform. Subsequently, instantaneous quality of service may be non-uniform in time.

In this context, Figure 4 shows a quality of service management diagram. Application 70 requests a particular quality of service to manager 72, which returns a

contract proposal. If accepted, the manager sends the parameters to the processing. The latter does runtime resource allocation (74), accounting and policing (76), and controls hardware elements 78-84. Resources so allocated render their services to application 70.

Figure 5 shows an exemplary resource requirement table that is present in the platform, inasmuch as the various streams, and consequently, the associated resource requirements have largely been standardized. Moreover, the application itself would not know the various resources present in the platform.

For MPEG video, the following objective quality levels for decoders may apply:

- 1. Black and white/half frame rate
- 2. Black and white/full frame rate
- 3. Full colour/half frame rate
- 4. Full colour/full frame rate

For MPEG audio, the following objective quality levels for decoders may apply:

- 1. Mono
- 2. Stereo
- 3. Dolby surround
- 4. 5+1 surround (5 standard channels plus one for special effects)

For 3-D graphics, the following objective quality levels for decoders may apply:

- 1. 5000 triangles or polygons per frame
- 2-4. 10k, 15k, and 20k triangles per frame, respectively.

In Figure 5, the columns list the resources required by these various quality levels from known facilities in the platform. The requirements as listed may pertain to averages, average plus a safety margin such as 20%, or to worst case. Herein, resources 1-4 may be first processor, second processor, Memory, and I/O facilities, respectively. Each resource is an abstract entity characterized by a maximum performance value.

In case the available resources extend beyond the bare minimum necessary for executing a particular application, the platform must decide as to which decoder should perform at what quality level. For this deciding, it is necessary to know the relative importance of a particular stream to the application, and also the necessary resources for its execution; the first knowledge is brought about by the application, inasmuch as it could widely vary between applications; the second is present in the platform itself. A straightfor-

ward way is to describe the executability levels in an incrementing manner. The term **decoder** is used for the actual decoding software package. For example:

Application: create MPEG2 video decoder 1 at quality level 1
create MPEG2 audio decoder 1 at quality level 1
raise MPEG2 audio decoder 1 to quality level 2
raise MPEG2 video decoder 1 to quality level 2
raise MPEG2 audio decoder 1 to quality level 3
player 1____
raise MPEG2 video decoder 1 to quality level 3
create MIDI-16 decoder 1 at quality level 1
player 2____
raise MIDI-16 decoder 1 to quality level 2
raise MIDI-16 decoder 1 to quality level 3

As shown in this table, for full functionality of the application, all three decoders should operate at quality level 3. For each of two exemplary available platforms, the actual performance is lower, and is given by the maximum level attained above the interrupted line for each of the decoders in question. For example, a relatively simple player has no MIDI, video at level 2, and audio at level 3. A relatively higher performing player has MIDI at level 1, and both audio and video at level 3. Obviously, the contribution by MIDI to the application is judged to be of relatively less importance. In another application, the relative importances of the streams could be different. For example, MIDI could be rated at approximately the same level of relevance for quality as audio. A different player could have a so-called 'sound blasters' card on board, which causes MIDI decoding taking place outside the central processor facilities.

In this context, Figure 6 shows a two-dimensional preference path diagram pertaining to a particular application, that has one dimension for each separate information stream of the application in question. For simplicity, a two-stream application has been taken in this example. The preference path is indicated as a solid line, and each cross therealong represents a sensible set of resources. Successive crosses indicate successive quality levels of playing the application. The preference path starts at the top left hand corner indicating zero resources. Subsequent crosses along the solid line correspond to successive rows in the same manner as given for the three-stream application considered earlier. Now, a particular cross (such as the one encircled) indicates minimum resources requirements for executing the

application on the platform in question. The platform tries to attain the best quality level, as represented by the farthest cross along the line of crosses.

Moreover, the dotted line indicates the full resources of the platform. The last cross **not** beyond this line indicates the maximum quality available by this particular platform. If the application has many streams in parallel, the matrix has as many dimensions, and moreover need not be static. In certain situations, the platform need not to keep strictly along the line of the diagram. For example, if MPEG-3 audio would just overtax the platform, the latter could still try whether MPEG-3 video (later in the sequence of steadily improving quality levels) could be feasibly executed by the platform. If yes, the latter may be included in the execution.

For the application, the specifying of the relative importance sequence may be a laborious task, especially if the number of streams is large and the assignment of relative importances is dynamic. Note that the number of streams may indeed be large, inasmuch as the number of languages may increase, the audio may be standard or karaoke, and also video may presented in more versions in parallel. To simplify this, a relative importance is assigned to each decoder, and these are mapped on an elementary model. Such is done as follows:

- The importance of a decoder is its minimally allowed quality level; this will corresponds to its first appearance in the incremental description.
- The quality of each decoder is incremented in the order in which they are opened.
- The quality of each decoder is incremented till it reaches maximum quality.

For example, an importance scheme with highest quality for each decoder at 3, and given as:

Decoder1 = OpenDecoder (MPEG2, Video,2)
Decoder2 = OpenDecoder (MPEG2, Audio,2)
Decoder3 = OpenDecoder (MIDI-16,1)
will give the following incremental description:
create MPEG2 video decoder 1 at quality level 2;
create MPEG2 audio decoder 1 at quality level 2;
create MIDI-16 decoder 1 at quality level 1;
raise MPEG2 video decoder 1 to quality level 3;
raise MPEG2 audio decoder 1 to quality level 3;
raise MIDI-16 decoder 1 to quality level 2;
raise MIDI-16 decoder 1 at quality level 3.

To select the quality level of each decoder, we start at the beginning of the incremental model. The decoders are created at a certain quality level and each decoder quality level is incremented following the preference path described in the incremental model. For each decoder that is created and for each quality level that is incremented, the resource requirements are looked up in the resource requirement table and added up to get the total resource requirements of the application. If the requirements for one of the resources exceeds its maximum value (i.e. 100%), no more decoder quality is implemented. This is the end of the preference path of this platform for the actual application or scene.

CLAIMS:

1. A method for storing multimedia application information for execution on different platforms, said information comprising a plurality of information streams for presentation in parallel to a user,

characterized in that for enabling execution scalability of said application, said method collectively assigns to said streams a single string of quality level items that along said string indicate stepwise increasing quality levels, in that each item specifies an associated decoding quality level for a particular one of said streams.

2. A method as claimed in Claim 1, wherein at least one such stream is associated to at least two non-contiguous items of said string, that are separated by at least one item pertaining to another stream.

3. A method as claimed in Claims 1 or 2, wherein the quality level items are assigned to said streams in a dynamic manner along said application when stored.

4. A method as claimed in Claims 1, 2 or 3, for a number of streams that is at least equal to three and to each of which are assigned at least two such quality levels.

5. A unitary physical record carrier containing a comprehensive multimedia application stored according to the method as claimed in any of Claims 1 to 4,

characterized by for enabling execution scalability of said application, said application has collectively assigned to said streams a single string of quality level items that along said string indicate a stepwise increasing quality levels, in that each item specifies an associated amount of associated decoding quality level for a particular one of said streams.

6. A record as claimed in Claim 5, wherein at least one such stream is associated to at least two non-contiguous items of said string, that are separated by at least one item pertaining to another stream.

7. A record as claimed in Claims 5 or 6, and comprising a plurality of said strings, each pertaining to a particular part of said application, and wherein the quality level items are assigned to said streams in a dynamic manner in said respective strings.

8. A record as claimed in Claims 5, 6 or 7, for a number of streams that is at least equal to three and to each of which are assigned at least two such quality levels.

9. A platform system for playing multimedia application information from a

storage medium, which application is executable on a range of different platforms and comprises a plurality of parallel information streams,

characterized in that for enabling execution scalability along said range, wherein said application has collectively assigned to said streams a single string of quality level items that along said string indicate stepwise increasing quality levels, in that each item specifies an associated quality level for a particular one of said streams,

said platform has interrogating means for interrogating said items along said string and accumulating all platform resources required for a particular quality level item, and comparing means fed by said interrogating means for comparing required platform resources to actual platform resources, and selecting means fed by said comparing means for selecting an optimum quality level as providable by the platform for the application in question.

10. A platform system as claimed in Claim 9, and having search means for searching additional string items beyond an item representing matching required platform resources to maximum available platform resources, whilst passing beyond an item overtaxing platform resources and pertaining to a first information stream, in favour of an item pertaining to a second information stream and commensurate with available platform resources, therewith improving said performance quality.

BNSDOCID: <WO__9715149A1_I_>

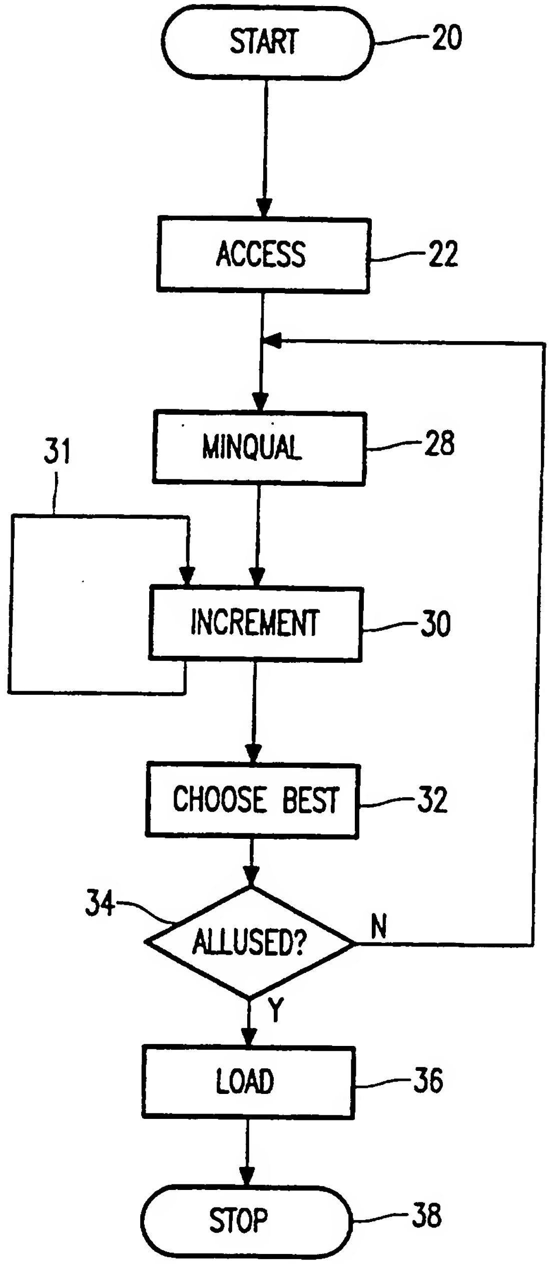

START
20
ACCESS
22
MINQUAL
28
31
INCREMENT
30
CHOOSE BEST
32
34
ALLUSED?
N
Y
LOAD
36
STOP
38

FIG. 1

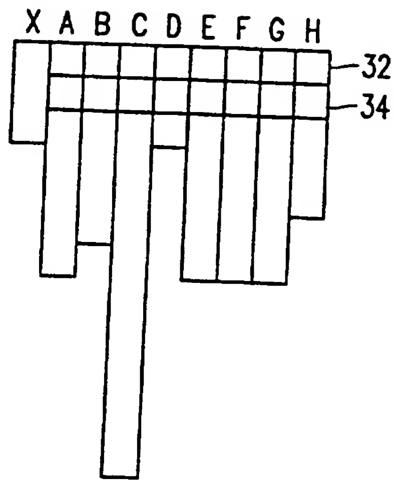


FIG. 2

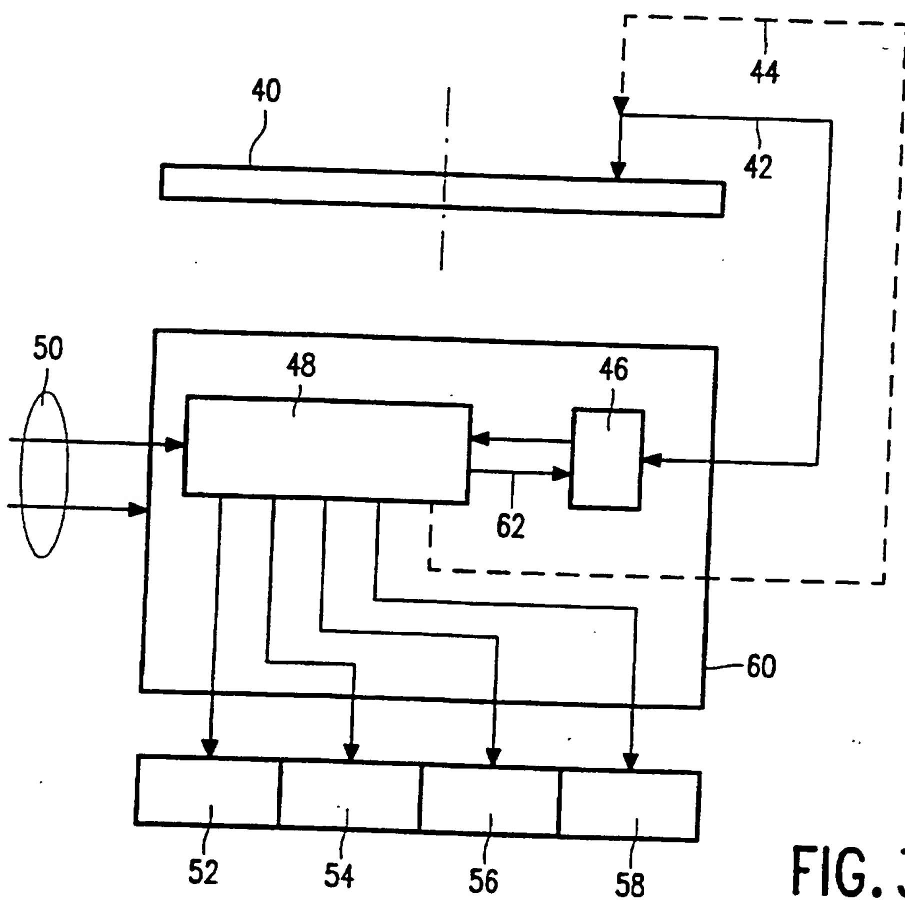


FIG. 3

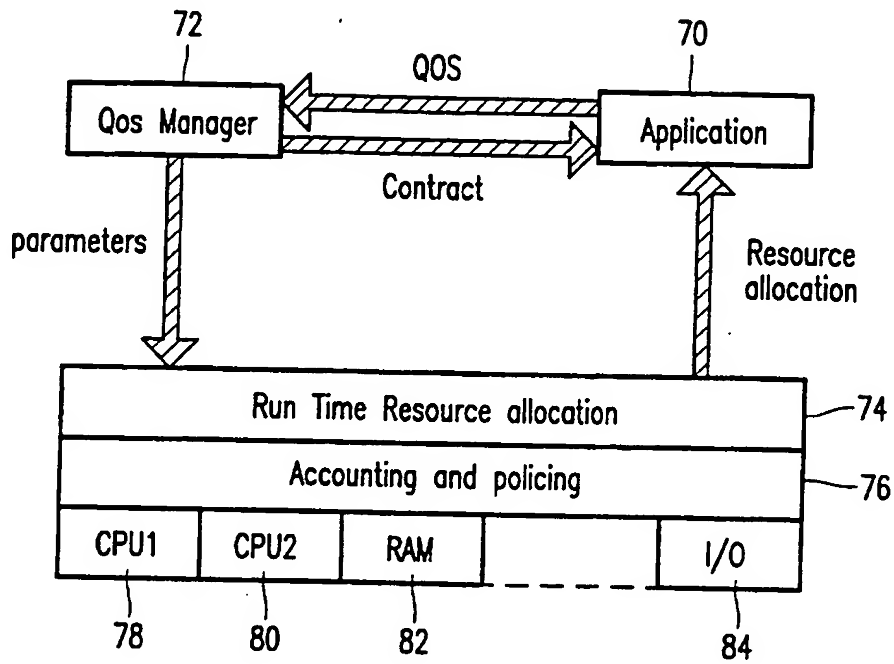

72
70
QOS
Qos Manager
Application
Contract
parameters
Resource allocation
Run Time Resource allocation
74
Accounting and policing
76
CPU1
CPU2
RAM
I/O
78
80
82
84

FIG. 4

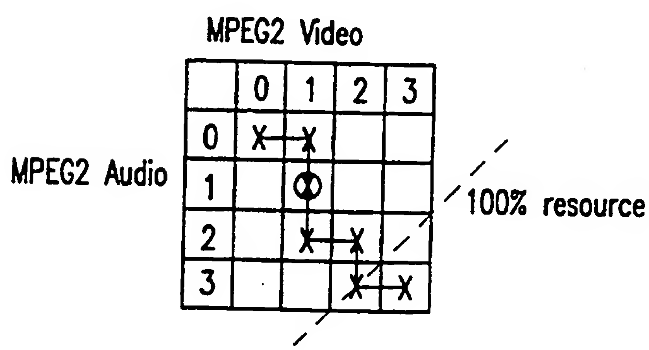

MPEG2 Video
0
1
2
3
MPEG2 Audio
0
1
2
3
100% resource

FIG. 6

An IMDF platform will have the following resource requirement table in memory

Decoder type	Resource 1	Resource 2	Resource 3	Resource 4
MPEG-video level 1	5%	20%	2 MB	0%
MPEG-video level 2	10%	20%	2 MB	0%
MPEG-video level 3	10%	20%	2 MB	0%
MPEG-video level 4	20%	20%	4 MB	0%
MPEG-audio level 1	...			
MPEG-audio level 2	...			
MPEG-audio level 3				
MPEG-audio level 4				
3D-Graphics level 1				
3D-Graphics level 1				
3D-Graphics level 1				
3D-Graphics level 1	25%	0%	4 MB	30%

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB 96/01074

A. CLASSIFICATION OF SUBJECT MATTER

IPC6: H04N 7/58

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC6: H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5196933 A (JEAN-PIERRE HENOT), 23 March 1993 (23.03.93), column 2, line 15 - column 4, line 25 --	1-10
A	US 5216503 A (WOO H.PAIK), 1 June 1993 (01.06.93), column 1, line 62 - column 3, line 24 --	1-10
A	US 5363138 A (HAYASHI ET AL), 8 November 1994 (08.11.94), column 1, line 49 - column 3, line 41 --	1-10
A	EP 0643537 A2 (PHILIPS ELECTRONICS N.V.), 6 Sept 1994 (06.09.94), page 2, line 1 - page 3, line 2 -- --------	1-10

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" erlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 March 1997	06.03.1997
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer KENNETH AHRENGART Telephone No. +46 8 782 25 00

BNSDOCID: <WO___9715149A1_I_>

INTERNATIONAL SEARCH REPORT

Information on patent family members

03/02/97

International application No.
PCT/IB 96/01074

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US-A- 5196933	23/03/93	DE-D,T- 69119634	23/01/97
		EP-A,B- 0448491	25/09/91
		FR-A,B- 2660139	27/09/91
US-A- 5216503	01/06/93	AU-B- 650219	09/06/94
		AU-A- 2995592	01/07/93
		CA-A- 2084178	25/06/93
		EP-A- 0550843	14/07/93
		JP-A- 6086261	25/03/94
US-A- 5363138	08/11/94	JP-A- 5122538	18/05/93
EP-A2- 0643537	06/09/94	AU-A- 7290294	23/03/95
		BE-A- 1007490	11/07/95
		CA-A- 2131573	11/03/95
		CN-A- 1111874	15/11/95
		JP-A- 7111649	25/04/95

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ______________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)